

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

***/АРХИТЕКТУРНО- АЛГОРИТМИЧЕСКИЕ
ОСНОВЫ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ/***

**Лекции для магистрантов второго года обучения.
Специализация 09.04.01**

***Лектор: профессор Райхлин
Вадим Абрамович***

ОСНОВНЫЕ ЦЕЛИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ :

- уяснить приоритетную роль параллельных вычислений в современных информационных технологиях, действительность и перспективу параллельных систем, принципы оценок их эффективности;
- создать базовые представления о принципах организации высокопроизводительных параллельных систем;
- осознать необходимость и освоить принципы использования библиотек параллельного программирования для выполнения высокопроизводительных вычислений.

ОСНОВНЫЕ ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ :

- знакомство со способами оценок производительности параллельных систем;
- показ на конкретных примерах адекватности параллельной обработки современным задачам информатики;
- знакомство с принципами организации основных классов современных параллельных компьютеров и суперпроцессоров, подсистем коммутации и памяти;
- изучение принципов работы библиотек параллельного программирования;
- овладение способами их использования;
- приобретение практических навыков решения ресурсоемких задач с применением библиотек параллельного программирования

ПРЕДМЕТ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ :

- архитектурные и программные основы высокопроизводительных параллельных вычислений.

МЕСТО ДИСЦИПЛИНЫ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Изучение дисциплины «Высокопроизводительные вычисления» предполагает наличие у студентов знаний по основам информатики и практических навыков программирования, полученных в процессе обучения в бакалавриате и специалитете.

Дисциплина «Высокопроизводительные вычисления» входит в вариативную часть профессионального цикла образовательной программы магистра. Материал курса основан на знаниях, навыках и умениях, почерпнутых студентами при обучении в магистратуре из курсов «Вычислительные системы», «Технология разработки программного обеспечения».

Полученные при изучении дисциплины знания, умения и навыки будут использованы студентами при прохождении научно-исследовательской практики и при подготовке выпускной квалификационной работы.

СТРУКТУРА ДИСЦИПЛИНЫ, ЕЕ ТРУДОЕМКОСТЬ

Наименование раздела и темы	Всего часов	Виды учебной деятельности, включая самостоятельную работу студентов и трудоемкость (в часах/ интерактивные часы)						Формы и вид контроля освоения
		лекции	лаб. раб.	пр. зан.	сам. раб.	контроль		
Раздел 1. Начальные понятия и предпосылки								
Тема 1.1. Необходимость, тенденции развития, классификация	7/2	2/2	—	—	5	—		Тесты, активность работы на лекциях
Тема 1.2. Показатели производи - тельности	7/2	2/2	—	—	5	—		
Тема 1..3.. Предметные предпосылки параллелизма	9/4	2/2	—	—	5	2/2		
Раздел 2. Представители параллельных систем								
Тема 2.1. Параллельные ассоциа- тивные процессоры	9/4	—	—	—	5	4/4		Тесты
Тема 2.2. Процессорные матрицы	9/4	—	—	—	5	4/4		
Тема 2.3. Мейнфреймовые архи тектуры и кластерные системы	7/2	2/2	—	—	5	—		Тесты, активность работы на лекциях
Тема 2.4. Сети коммутации	9/4	1/1	—	—	5	3/3		
Раздел 3. Память, суперпроцессоры								
Тема 3.1. Организация главной памяти	5,5/0,5	0,5/0,5	—	—	5	—		Тесты, активность работы на лекциях
Тема 3.2. RAID-массивы	5,5/0,5	0,5/0,5	—	—	5	—		
Тема 3.3. Суперпроцессоры	10/5	2/2	—	—	5	3/3		

Раздел 4. Данные и задачи, пакеты для разработки параллельных программ								
Тема 4.1. Разделяемые данные. Пакеты MPICH и MPI.NET. Стандарт OpenMP	11/5	–	4/3	–	5	2/2		Тесты, активность работы на ЛР 1-2, защита ЛР 1-2
Тема 4.2. Управление задачами	11/5	–	4/3	–	5	2/2		
Раздел 5. Параллельные циклы и запросы, программирование видеокарт								
Тема 5.1. Параллельные циклы. Про- граммирование видеокарт nVidia	11/5	–	4/3	–	5	2/2		Тесты, активность работы на ЛР3-4, защита ЛР 3-4
Тема 5.2. Параллельный LINQ	11/5	–	4/3	–	5	2/2		
Раздел 6. Прикладные библиотеки NVIDIA CUDA								
Тема 6.1. Линейная алгебра с плот- ными и разреженными матрицами	11/5	–	4/3	–	5	2/2		Тесты, активность работы на ЛР5-6, защита ЛР 5-6
Тема 6.2. Преобразование Фурье. Генерация псевдослучайных чисел	11/5	–	4/3	–	5	2/2		
Раздел 7. Библиотека шаблонов Thrust								
Тема 7.1. Контейнеры, итераторы, алгоритмы	9/4	–	–	–	5	4/4		Тесты
Тема 7.2. Типы трансформаций, zip- итераторы	9/4	–	–	–	5	4/4		
Курсовая работа/проект	–	–	–	–	–	–	–	–
Экзамен	18	–	–	–	18	–		
ИТОГО:	180/66	12/12	24/18	–	108	36/36		

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

- **Райхлин В.А.** Системы параллельной обработки данных. – Казань: Изд-во ФЭН, 2010.
- **Райхлин В.А.** Презентация лекций по ПВ http:
[//modelling.kai.ru/pv.zip](http://modelling.kai.ru/pv.zip)
=====
- **Райхлин В.А.** Начала параллельных вычислений. *Материалы лекций.* – Казань: Изд-во КГТУ, 2008. http:
[//modelling.kai.ru/LPC.zip](http://modelling.kai.ru/LPC.zip)
- **Райхлин В.А.** Суперпроцессоры и RAID-массивы. *Материалы к лекциям по параллельным вычислениям.* –
http://modelling.kai.ru/SP_raid.zip
=====
- **Воеводин В.В., Воеводин Вл.В.** Параллельные вычисления. – С.-Пб.: “БХВ-Петербург”, 2004.
- **Корнеев В.В.** Вычислительные системы. – М.: «Гелиос АРВ», 2004.

Лекции 1-3

НАЧАЛЬНЫЕ ПОНЯТИЯ И ПРЕДПОСЫЛКИ

Лекция 1. НЕОБХОДИМОСТЬ, ТЕНДЕНЦИИ РАЗВИТИЯ, КЛАССИФИКАЦИЯ

1. ОБЩИЕ ПОЛОЖЕНИЯ

СФЕРА ПРИМЕНЕНИЙ ВВС

- **В коммерции** – работы со сверхбольшими базами данных и графическими приложениями: кино, телевидение.
- **В военных целях** – разработка ядерного оружия, конструирование самолетов, ракет, бесшумных подводных лодок.
- **В научных исследованиях** – физических, химических, метеорологических, геологических.
- **В технике** – решение проблем аэрокосмической, автомобильной, газовой и нефтедобывающей промышленности, связанных со своевременной обработкой больших объемов экспериментальных данных.

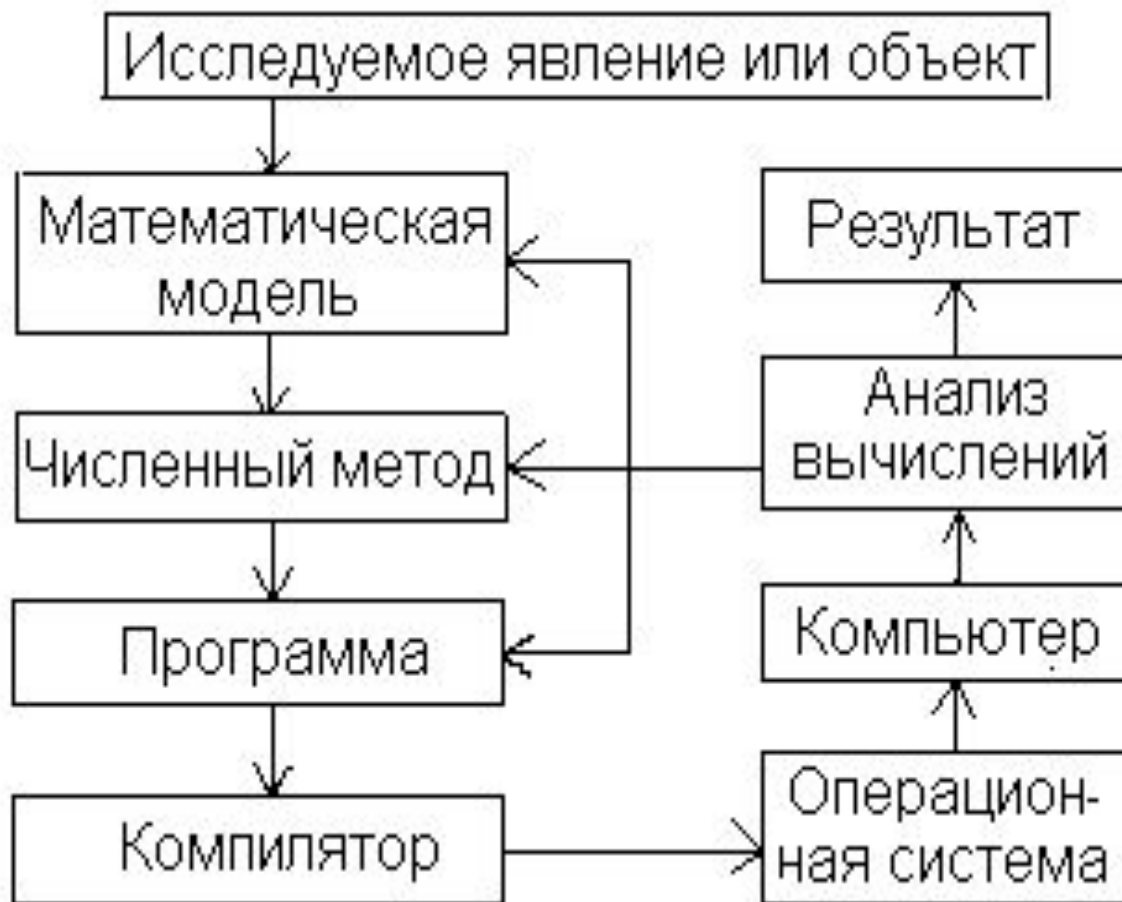
ОСНОВА ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ВС

введение параллелизма на всех уровнях: алгоритмическом, программном, структурном, архитектурном.

ПРИМЕНЕНИЕ ВВС СВЯЗЫВАЕТСЯ

с проведением сложного компьютерного моделирования (вычислительного эксперимента).

ЭТАПЫ ЧИСЛЕННОГО ЭКСПЕРИМЕНТА



ВВЕДЕНИЕ ПАРАЛЛЕЛИЗМА В АРХИТЕКТУРУ ЭВМ

- **КОНВЕЙЕРНАЯ ОБРАБОТКА** – применение метода линии сборки с целью повышения производительности арифметического и управляющего устройств.
- **ФУНКЦИОНАЛЬНАЯ ОБРАБОТКА** – предоставление нескольким независимым устройствам возможности выполнения различных функций, таких как операции логики, сложения или умножения, обеспечивая взаимодействие с различными данными.
- **МАТРИЧНАЯ ОБРАБОТКА** – наличие матрицы идентичных процессорных элементов с общей системой управления, где все элементы в каждый момент времени выполняют одну и ту же операцию, но с разными данными, хранящимися в их собственной либо в общей памяти.
- **МУЛЬТИПРОЦЕССОРНАЯ ОБРАБОТКА** – осуществляется множеством процессоров, каждый из которых выполняет свои собственные команды, а все процессоры взаимодействуют друг с другом тем или иным способом.

РЕТРОСПЕКТИВА РАЗВИТИЯ ПАРАЛЛЕЛЬНЫХ СИСТЕМ

а) Совершенствование архитектуры фон Неймана

- Быстродействующие скалярные компьютеры.
- Конвейерные векторные ЭВМ.
- Алгоритмические матричные процессоры.

б) Переход к новым параллельным архитектурам

- Процессорные матрицы.
- Ортогональные и ассоциативные ЭВМ.

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ

- Векторно-конвейерные суперкомпьютеры.
- SMP-системы.
- NUMA-технологии.
- MPP-системы.
- кластерные системы.

2. СИСТЕМАТИКА ПАРАЛЛЕЛЬНЫХ СИСТЕМ

ЗАДАЧИ СИСТЕМАТИКИ

- Однозначное отнесение любой известной или предвидимой архитектуры к тому или иному классу (**«что есть что»**).
- Выделение существенных различий между элементами одного класса (**морфологический анализ**).

СИСТЕМАТИКА ФЛИННА

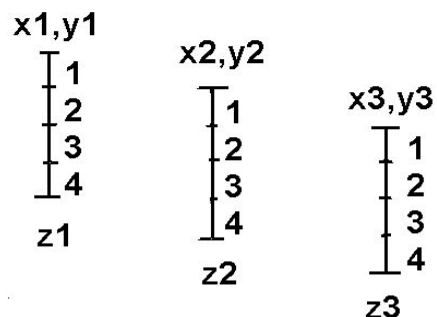
- ОКОД (SISD) – **один поток команд/один поток данных** – как в ЭВМ **фон Неймана**. Команда > одна скалярная операция над одним (парой) операндов.
- ОКМД (SIMD) – **один поток команд/много потоков данных**. Здесь сохраняется один поток команд, но уже векторных, которые инициируют множество операций. Каждый элемент вектора рассматривается как элемент отдельного потока данных. Этот класс включает все **машины с векторными командами**.
- МКОД (MISD) – **много потоков команд/один поток данных**. Этот класс **пока вакантен**, т. к. здесь несколько команд должны работать с одним элементом данных.
- МКМД (MIMD) – **много потоков команд/много потоков данных**. Этот класс включает в себя все формы **мультипроцессорных конфигураций**.

СТРУКТУРНАЯ СИСТЕМАТИКА

– решает задачи **морфологического анализа**.

Лекция 2. ПОКАЗАТЕЛИ ПРОИЗВОДИТЕЛЬНОСТИ

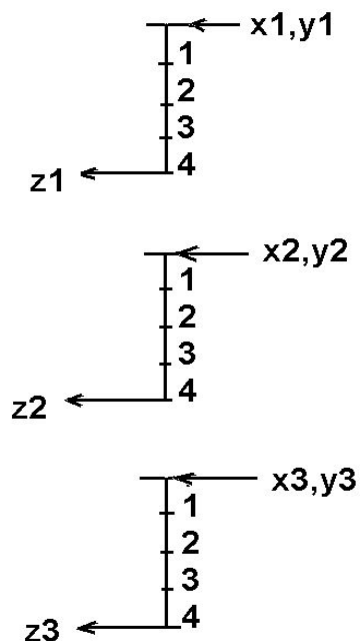
КОНВЕЙЕРНАЯ



1 ТАКТ / РЕЗУЛЬТАТ

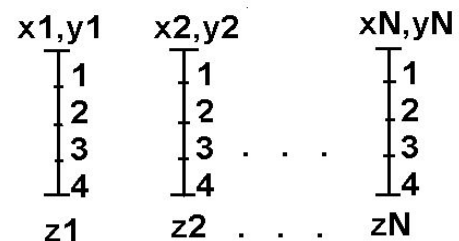
STAR - 100
CRAY - 1
CYBER - 205

ПОСЛЕДОВАТЕЛЬНАЯ



4 ТАКТА / РЕЗУЛЬТАТ

МАТРИЧНАЯ



4 ТАКТА / N РЕЗУЛЬТАТОВ

ILLIAC IV
ICL DAP
STARAN

а) АБСТРАКТНЫЕ ОЦЕНКИ

- **Время выполнения векторной арифметической операции**

$$t = b + cn.$$

Или

$$t = c \left(n + \frac{b}{c} \right) = r_{\infty}^{-1} (n + n_{1/2})$$

$$r_{\infty} = \frac{1}{c} = \lim_{n \rightarrow \infty} \frac{n}{t}$$

— **пиковая**, или асимптотическая **производительность**. Измеряется числом эквивалентных скалярных операций в секунду (**MFLOPS**).

$$n_{1/2} = b/c$$

— **длина полупроизводительности**, т.е. длина вектора, при которой достигается половина пиковой производительности.

Оценки векторной производительности

КОМПЬЮТЕР	$n_{1/2}$	r_{∞} MFLOPS
64' CRAY-1	10–20	80
48' BSP	25–150	50
2-конв. CDC CYBER 205	100	100
64' CDC STAR 100	150	25
32' (64x64) ICL DAP	2048	16

СИСТЕМНАЯ ПРОИЗВОДИТЕЛЬНОСТЬ

ОСНОВНЫЕ НЕДОСТАТКИ АБСТРАКТНОЙ ОЦЕНКИ:

1. Игнорирование времени на выполнение скалярных операций– организацию циклов, переходов и др.

Ускорение S для N процессоров (закон Амдала)

$$S \leq \frac{1}{f + \frac{1-f}{N}}$$

f – доля трудозатрат на выполнение скалярных операций. Для получения заданного S , требуется разработка специальных параллельных алгоритмов с $f < 1/S$.

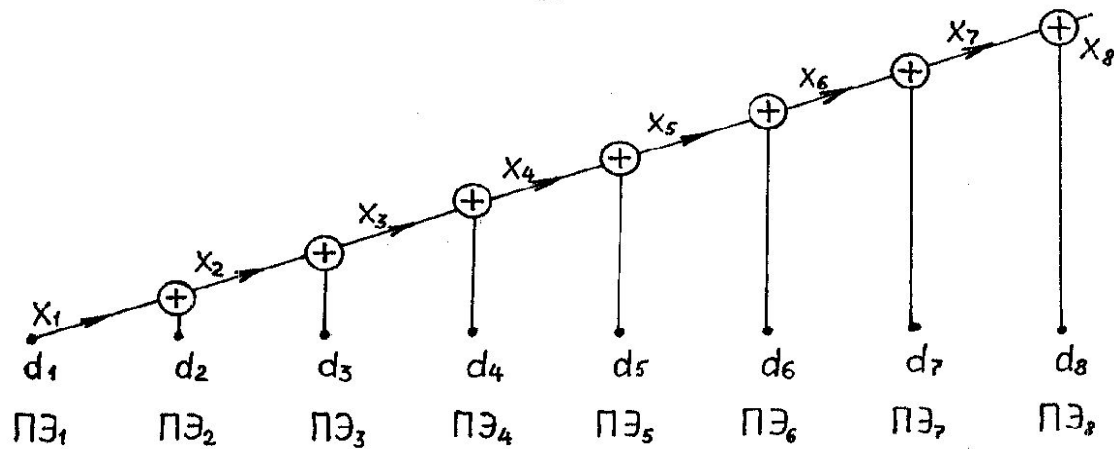
2. Игнорирование потерь на маршрутизацию.

Время маршрутизации (пересылки данных между процессорами) с ростом N может доминировать.

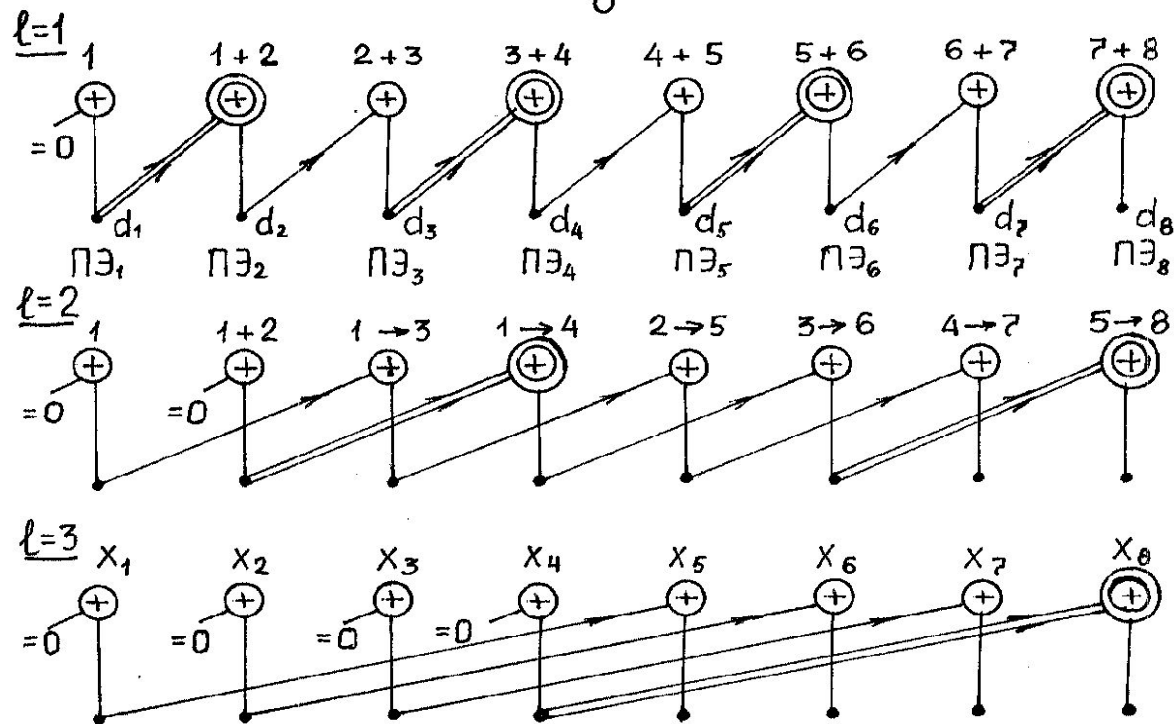
$$x_j = \sum_{k=1}^j d_k, \quad j = \overline{1, n}$$

Пример. Вычисление последовательных сумм можно реализовать как $(n - 1)$ сложений: $x_j = x_{j-1} + d_j, x_0 = 0$.

a



b



На l -уровне выполняется сдвиг на 2^l позиций. Уровень $l = \log_2 n$ дает искомую сумму в крайнем правом аккумуляторе. Так что число операций суммирования составляет теперь уже

$$\log_2 n < n - 1.$$

В случае связи “к ближайшему соседу” на уровне l потребуется 2^{l-1} единичных операций маршрутизации, что дает в целом

$$1 + 2 + \dots + n/2 = n - 1$$

таких операций со степенью параллелизма ‘ n ’.

Пусть r – отношение времени выполнения одной операции суммирования ко времени единичной операции маршрутизации. Тогда отношение суммарных времен равно

$$r \log_2 n / (n-1),$$

т.е. с ростом ‘ n ’ влияние маршрутизации растет.

При переходе от рекурсии к каскадному суммированию число эквивалентных скалярных арифметических операций растет с $(n - 1)$ до $n \log_2 n$.

Поэтому *эмуляция параллельного алгоритма на последовательной ЭВМ будет всегда малоэффективной.*

б) ОТНОСИТЕЛЬНАЯ ОЦЕНКА СИСТЕМНОЙ ПРОИЗВОДИТЕЛЬНОСТИ

ПОЛАГАЕТСЯ:

- Параллельный процессор (ПП) – *внешний акселератор Host.*
- Длительности тактов ПП и Host *одинаковы.*

Для каждого элемента множества **W** *представительных программ* находятся *модельные оценки производительности Host и комплекса в целом. Усредненный коэффициент ускорения S* на множестве **W** *дает искомую оценку. Условие равенства длительностей тактов делает такую оценку* **технологически независимой.**

Эффективность комплекса $H = S/D$, $D > 1$ – *коэффициент роста аппаратных затрат комплекса в сравнении с Host.* Чем выше **H**, тем *перспективнее система* для данной предметной области.

Ускорение выполнения матричных команд и подпрограмм обработки массивов 32x32 бит

Матричные команды и подпрограммы ПМА	$\frac{t_{Host}}{t_{MA}}$	Матричные команды и подпрограммы ПМА	$\frac{t_{Host}}{t_{MA}}$
1. Структурная обработка		3. Логическая обработка	
Поиск на "=", "≠", ">"	535	V, &, ⊗: массовые	557
Сортировка	84,3	редуктивные	114
Транспонирование	143	Логическое умножение булевых матриц	159
2. Симеольная обработка		4. Арифметическая обработка	
Удаление символа со сжатием	263	Сложение кодов: массовое	446
Перекодировка	149	редуктивное	158
Выравнивание по границе	157	Арифметика с плавающей точкой :	
Условная запись	1046	Сложение массовое	19
		редуктивное	13
		Умножение массовое	19

Ускорение $t_{\text{HOST}}/t_{\text{HOST-ПМА}}$ на множестве представительных программ

№ п/п	ПРЕДСТАВИТЕЛЬНЫЕ ПРОГРАММЫ	$t_{\text{Host}} / t_{\text{Host-ПМА}}$
1	Транспонирование булевой матрицы 512 x 320 бит	26,3
2	Сортировка массива 4096 32-разрядных кодов	16
3	Минимизация СБФ, заданной на 64 интервалах. Число функций (переменных) – не более 32	8,4
4	Решение уравнения Пуассона, область – 16 000 точек	16,4
5	Комплексное БПФ с плавающей точкой, N = 1024	10
6	Распознавание двоичных образов 16 x 16 байт в кадре 112 x 128 байт	10,5 – 12,8

В) ТЕСТОВЫЕ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ

Эти оценки наиболее объективны. Разработкой и стандартизацией тестов оценки производительности современных компьютеров занимается ряд корпораций и комитетов (советов): **SPEC, LINPAC, AIM, TPC** и др.

Специальные тесты для определения производительности:

- только *процессора* (пример **LINPACK**),
- только *файловой системы* (пример **Bonnie**),
- только *коммутационной сети* (пример **MPI-тестов**),
- *комбинированные* тесты (пример **AIM**).

По результатам **LINPAC**-тестирования ведется список **TOP-500**, в котором ежегодно ранжируются **500** наиболее производительных в мире систем.

Тесты **TPC A, B, C, D, E** (*TPC – Transaction Processing Performance Council* – Совет по производительности обработки транзакций, основан в 1988г.) разработаны специально для оценки производительности **СУБД**.

Лекция 3. ПРЕДМЕТНЫЕ ПРЕДПОСЫЛКИ

1. ОБРАБОТКА СИГНАЛОВ

(преобразов-е формы или частот. спектра эл., речевых, видео сигналов)

Электрические и речевые сигналы – некоторые временные функции $f(t)$.
Время t и значения $f(t)$ непрерывны – аналоговые величины.

Сигналы изображения – функции $f(X,Y)$ или $f(X,Y,Z)$ изменения цветности, яркости и др. в 2- или 3-мерной системе координат.

Цифровая обработка сигналов (ЦОС) широко применяется при передаче речевых сигналов и сигналов изображений, при распознавании и синтезе речи, используется в медицине, метеорологии, сейсмологии и др.

Исходные аналоговые сигналы путем *дискретизации* во времени и *квантования* по амплитуде преобразуются в последовательность цифровых данных. Далее она подвергается обработке, основанной на

- *дискретном преобразовании Фурье (ДПФ),*
- *быстром преобразовании Фурье (БПФ),*
- *преобразовании Адамара и др.,*
- *преобразованиях частотного спектра (цифровая фильтрация).*

Дискретизация во времени – представление $f(t)$ как последовательности значений $f(nT)$ в моменты времени, кратные интервалу T . Если частотный спектр $f(t)$ ограничен значением F , то $T = 1/(2F)$ – теорема Котельникова.

Дискретизация в пространстве – вопрос сложный.

Квантование – представление $f(nT)$ n -разр. двоич. числом с макс. ошибкой квантования $1/2^n$, **$n = 12 \dots 18$** (эл. сигналы), **$8 \dots 14$** (речь), **$4 \dots 9$** (изображ.)

ДИСКРЕТНОЕ И БЫСТРОЕ ПРЕОБРАЗОВАНИЯ ФУРЬЕ

***ДПФ, БПФ, Алгоритм Кули-Тьюки –
ПРОРАБОТАТЬ САМОСТОЯТЕЛЬНО !***

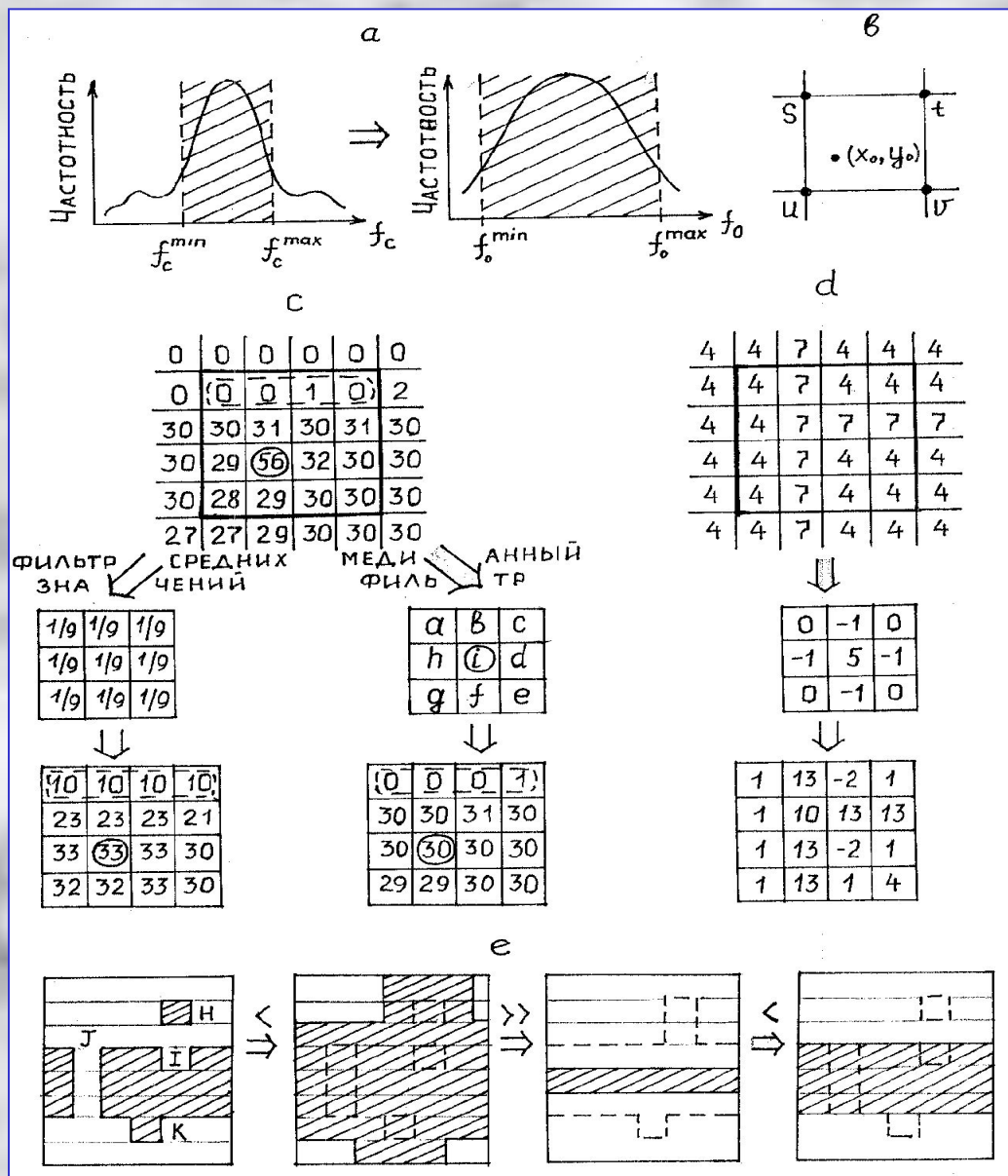
2. ОБРАБОТКА ИЗОБРАЖЕНИЙ

- **Обработка изображений** — выполнение различных операций над *многомерными сигналами*: *телевизионные изображения, чертежи и рисунки, фотографии* разведывательного характера, медицинские *рентгенограммы*, *электронно-микроскопические* фотографии молекул, *радио- и звуко- локаторные* карты, диаграммы *сейсмических* данных и др.
- **Основные виды обработки** — *улучшение изображений, их эффективное кодирование, распознавание образов, машинная графика.*
- **Области применения** — *медицина, дистанционное зондирование, идентификация личности, промышленные измерения, информационной служба* и т.д.

а) ОСНОВНЫЕ ПОНЯТИЯ

- **ДАННЫЕ ИЗОБРАЖЕНИЯ – ПИКСЕЛИ** – элементы двумерного *тхп* массива – бинарные (2 градации), многоградационные (например, 256 градаций) или многоградационно-векторные (256 градаций по каждой из составляющих – *красной, зеленой и синей*). Соответственно изображение – бинарное, полутонное или спектральное. Обычно *т, п* = 256...512 (иногда – до 10^7 и $>$). Одинаковые операции выполняются *параллельно* по всему изображению, что адекватно использованию *процессорных матриц*. Часто применяют *специальные графические приставки* к ПК.
- **УЛУЧШЕНИЕ (РЕСТАВРАЦИЯ) ИЗОБРАЖЕНИЙ** – компенсация искажений, вносимых при их формировании системами отображения.
- **КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ** – сокращение числа битов представления изображений, при достоверности их воспроизведения. Сначала – преобразование изображения. Затем – кодирование результата преобразования
- **РАСПОЗНАВАНИЕ ОБРАЗОВ** – это и распознаванию знаков, и средство медицинской диагностики, и составление карт земных ресурсов на основе фотографий, получен. со спутников (*дистанционное зондирование*), и др.
- **МАШИННАЯ ГРАФИКА** – ввод графич. информации (чертежей и рис.) в ЭВМ, ее обработка и вывод. Основная задача – *синтез и представление изображения*. Области применения: *компьютерная мультипликация, машинное проектир. логич. схем, выполнение дизайнерских проектов* и др.

б) УЛУЧШЕНИЕ ИЗОБРАЖЕНИЙ



Улучшение контрастности изображения дает градацион. преобразование

$$\mathbf{f}_0 = \mathbf{A} \mathbf{f}_c + \mathbf{B},$$

где $\mathbf{f}_c$ и $\mathbf{f}_0$ – значения видеоданных до и после преобразования (рис. *a*). Коэффициенты $\mathbf{A}$ и $\mathbf{B}$ определяют из условий перехода

$$\mathbf{f}_c^{\max} \rightarrow \mathbf{f}_0^{\max}, \quad \mathbf{f}_c^{\min} \rightarrow \mathbf{f}_0^{\min}.$$

Коррекция геометрических искажений (повороты или параллельные перемещения элементов) достигается преобразованием координат

$$\begin{vmatrix} x_0 \\ y_0 \end{vmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \begin{vmatrix} x_c \\ y_c \end{vmatrix}$$

Значения параметров a, b, c, d определяются по степени искажений некоторых хорошо известных элементов. Процедура такова:

$$(x_c, y_c) \rightarrow (x_0, y_0); \quad \mathbf{f}_0(x_0, y_0) := \mathbf{f}_c(x_c, y_c).$$

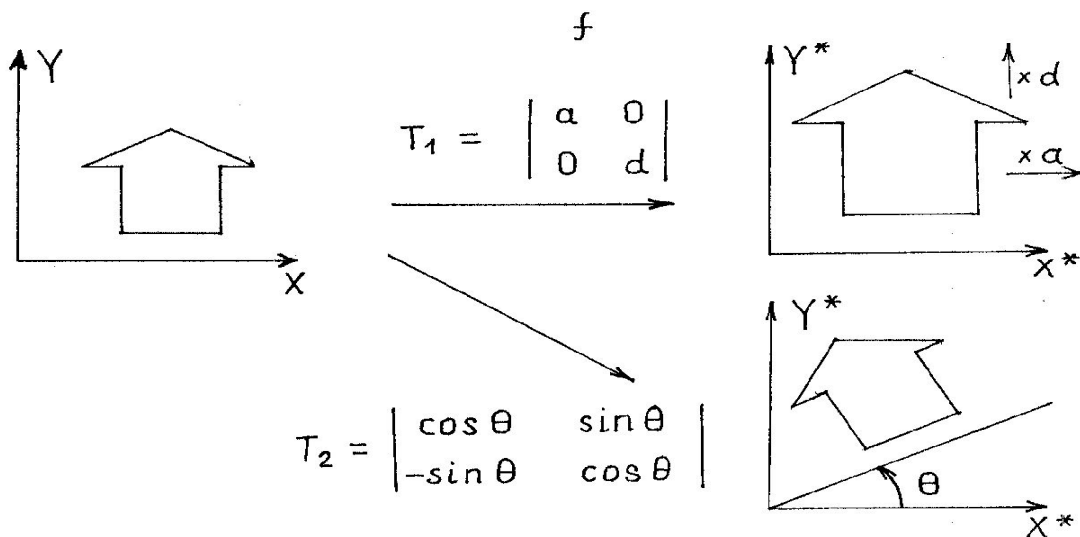
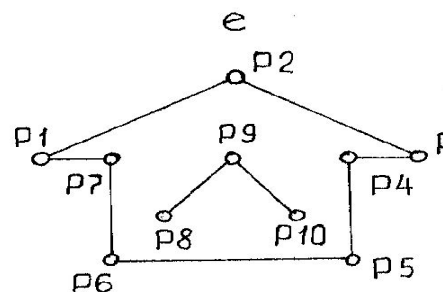
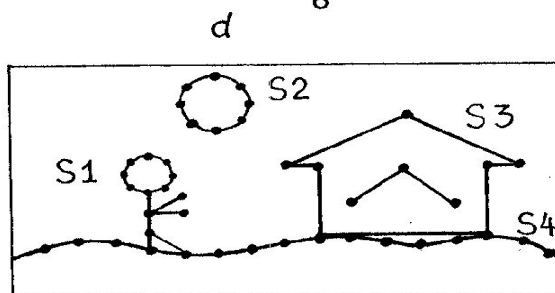
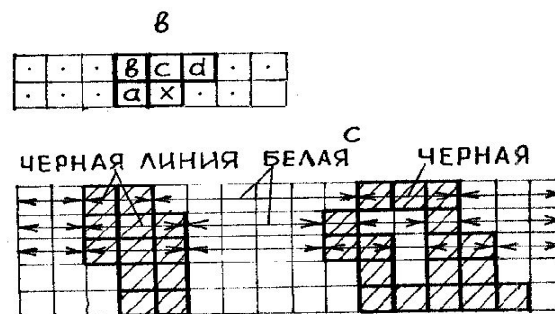
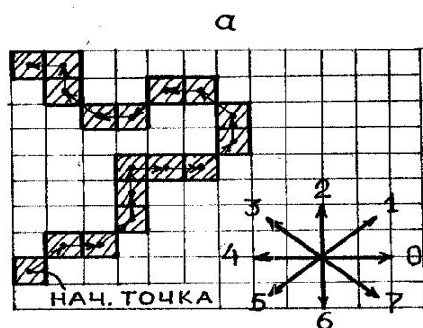
Дробные координаты (x_0, y_0) округляют до ближайшей овокупности (рис. *b*), принимая

$$\mathbf{f}_0(\mathbf{u}) := \mathbf{f}_0(x_0, y_0).$$

Типичные шумы изображения – зернистый шум и пятна на полутоновом изображении, отдельные шумы и обрывы линий на бинарном изображении. Обычно эти шумы могут быть устранены проведением для *каждого элемента изображения* локальной фильтрации на окружающем его участке 3×3 элементов, центром которого он является (рис. *c, d, e*).

Для эффективной обработки изображений целесообразно применение процессорных матриц со связью между 8 соседями.

В) КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ И ОБРАБОТКА ГРАФИКИ



ОБРАБОТКА СИМВОЛОВ

- **Обработка символов** – связана с редактированием текстов, переводом с одного языка на другой, доказательством теорем, преобразованием математических формул, медицинской диагностикой и т.д. В целом – с созданием **искусственного интеллекта**.

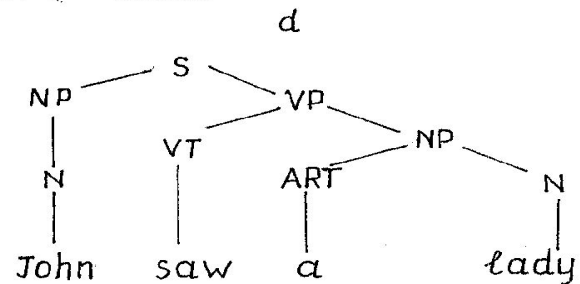
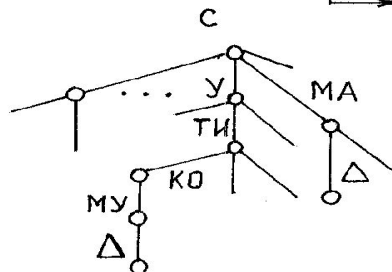
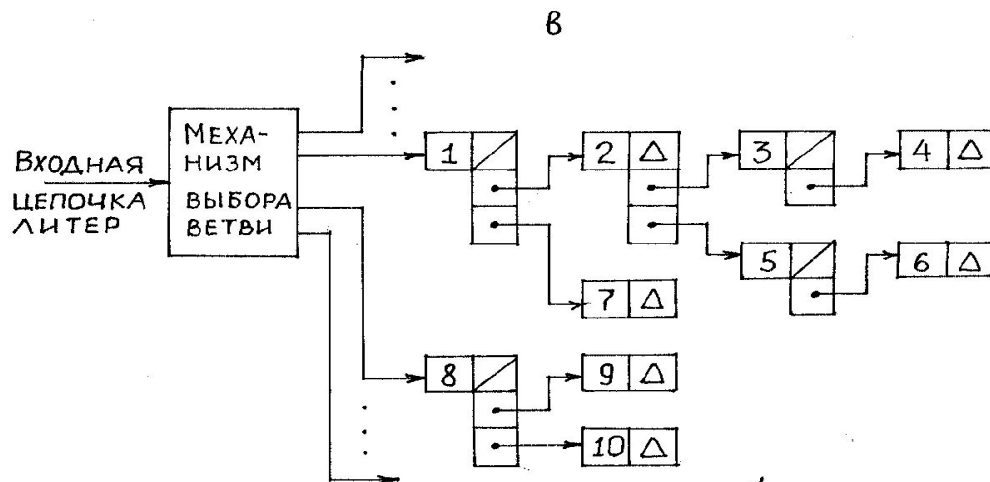
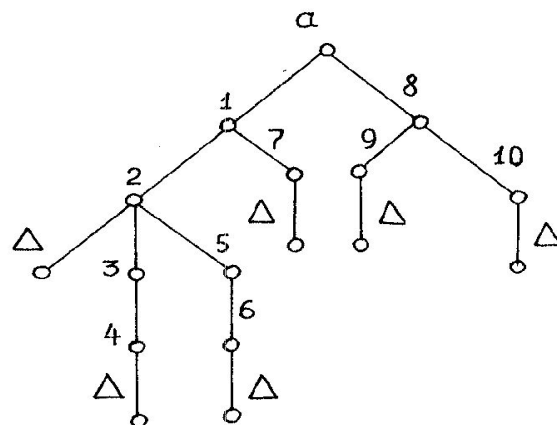
ОБРАБОТКА ЦЕПОЧЕК СИМВОЛОВ

- конкатенация (объединение нескольких цепочек),
сопоставление (сравнение двух цепочек),
замещение (замена одной цепочки на другую),
– выборка (выборка части цепочки).
- **Конкатенация:** $Z = XY$ либо $Z = X \text{ '}' Y$
- **Сопоставление:** (СРАВНИВАЕМАЯ ЦЕПОЧКА) (ОБРАЗЦОВАЯ ЦЕПОЧКА)
В наихудшем случае – $n(m-n)$ сравнений, m и n ($\leq m$) – длины сравниваемой и образцовой цепочек.
Алгоритм
КМП – развит Кнудом, Моррисом и Праттом.
- **Замещение:** $ZY = \text{'mosq'}$; $Z = \text{'knpt' '}' \text{'alsvi'}$, $Y = \text{'alsvi'}$.
 $Z = \text{'knpt' '}' \text{'mosq'}$
- **Операцию сравнения последовательностей литер целесообразно распараллелить.**

ОБРАБОТКА ЕСТЕСТВЕННЫХ ЯЗЫКОВ

- **ЕСТЕСТВЕННЫЙ ЯЗЫК** – используемый в повседневной жизни.
- **ВИДЫ ОБРАБОТКИ ЕЯ :**
 - обработка слов (поиск в словаре, обработка морфем);
 - обработка предложений (синтаксическая, семантическая);
 - обработка текстов (обработка контекста).
- **ТЕРМИНОЛОГИЯ:**
 - Слово** – последовательность букв.
 - Словарь** – все слова данного текста должны находиться в словаре для этого текста.
 - Предложение** – ряд нескольких слов.
 - Морфема** – наименьшая языковая единица: слово, префикс, суффикс.
- **ПОИСК В СЛОВАРЕ, ОРГАНИЗОВАННОМ КАК TRIE-ДЕРЕВО**
 - Пример:** {1 2, 1 2 3 4, 1 2 5 6, 1 7, 8 9, 8 10}
 - Механизм выбора последовательности узлов при поиске.**
 - Обработка морфем.**

ОБРАБОТКА СЛОВ И СИНТАКСИЧЕСКАЯ ОБРАБОТКА



СИНТАКСИЧЕСКАЯ ОБРАБОТКА, или ГРАММАТИЧЕСКИЙ РАЗБОР

- **СТРУКТУРА ПРЕДЛОЖЕНИЯ** определяет связи между объектами:
- **S** – предложение,
- **N** – существительное,
- **VT** – глагол,
- **ART** – артикль,
- **PREP** – предлог,
- **NP** – существительное или существительное с артиклем,
- **VP** – глагол и существительное (без или с артиклем) или существительное с артиклем и с предлогом,
- **PP** – существительное (без или с артиклем) с предлогом.
- **ПРАВИЛА CFG** (context-free grammar – контекстно-свободная грамматика) для структуры предложения в английском языке:
- **S NP VP;**
- **NP ART N** либо **NP N;**
- **VP VT NP** либо **VP VT PP;**
- **PP PREP NP.**
- Согласно этим правилам, для предложения **‘John saw a lady’** получаем дерево грамматического разбора рис.d.